AFRL-AFOSR-JP-TR-2016-0055

Registration of Large Motion Blurred Images

Ambasamudram Rajagopalan
INDIAN INSTITUTE OF TECHNOLOGY MADRAS

05/09/2016
Final Report

Air Force Research Laboratory
AF Office Of Scientific Research (AFOSR)/ IOA
Arlington, Virginia 22203
Air Force Materiel Command

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
09-05-2016	Final	30 Sep 2014 to 29 Sep 2015

4. TITLE AND SUBTITLE
Registration of Large Motion Blurred Images

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA2386-14-1-4092

5c. PROGRAM ELEMENT NUMBER
61102F

6. AUTHOR(S)
Ambasamudram Rajagopalan

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
INDIAN INSTITUTE OF TECHNOLOGY MADRAS
SARDAR PATEL ROAD
Chennai, 600036 IN

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AOARD
UNIT 45002
APO AP 96338-5002

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/AFOSR IOA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-AFOSR-JP-TR-2016-0055

12. DISTRIBUTION/AVAILABILITY STATEMENT
A DISTRIBUTION UNLIMITED: PB Public Release

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Registration and change detection occupy an important place in aerial surveillance in which the goal is to mark regions of changes such as objects, buildings etc. over time. The challenge lies in handling the dynamics of the capturing system, for example, a drone. CMOS sensors, used in recent times, when employed in these cameras produce two types of distortions, namely the rolling shutter effect and motion blur. These are intertwined due to sequential row exposure mechanism. In this report, we propose a layered registration procedure to address the problem of detecting changes in 3D scenes affected by motion blur and rolling shutter artifacts. The proposed method has been tested on synthetic as well as real data.

15. SUBJECT TERMS
Image Processing, Motion Blur

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			HONG, SENG
Unclassified	Unclassified	Unclassified	SAR	18	**19b. TELEPHONE NUMBER** *(Include area code)* 315-229-3519

FINAL REPORT FOR AOARD-AFRL

(FA2386-14-1-4092, Year 2014-2015)

TITLE	**Registration of Large Motion Blurred Images**
PI	Prof. A.N. Rajagopalan, Indian Institute of Technology Madras
AFRL POC	Dr. Guna Seetharaman, Civ DR-IV AFRL/RIEA
AOARD PM	Dr. Seng Hong
AFOSR PM	Dr. Frederica Darema, AFOSR/RSL
Duration	Oct. 1. 2014 - September 30, 2015

Abstract

Registration and change detection occupy an important place in aerial surveillance in which the goal is to mark regions of changes such as objects, buildings etc. over time. The challenge lies in handling the dynamics of the capturing system, for example, a drone. CMOS sensors, used in recent times, when employed in these cameras produce two types of distortions, namely the rolling shutter effect and motion blur. These are intertwined due to sequential row exposure mechanism. In this report, we propose a layered registration procedure to address the problem of detecting changes in 3D scenes affected by motion blur and rolling shutter artifacts. The proposed method has been tested on synthetic as well as real data.

1 Introduction

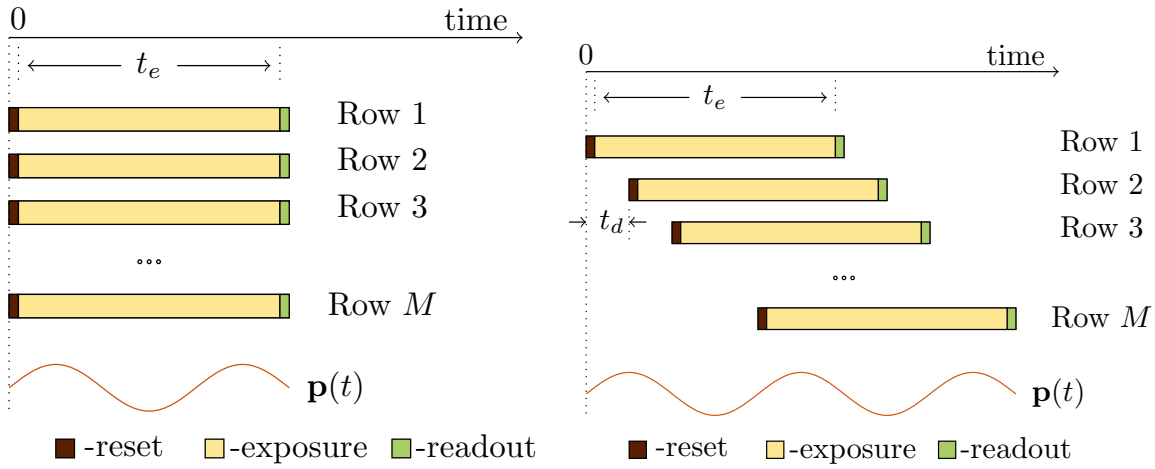
The problem of registration and change detection in images is a highly researched topic in image processing and computer vision due to its ubiquitous use in a wide range of domains including surveillance, (Feris et al. [1]) tracking, (Sivaraman et al. [2]) driver assistance systems (Fang et al. [3]) and remote sensing (Bruzzone and Bovolo [4]). The related work of background modeling for video frames has also been well-studied (Bouwmans [5]). The aim of this class of methods is to learn background regions from the past frames and detect objects in the current frame. Examples include statistical models such as mixture of Gaussians, clustering models like K-means algorithm, wavelet modeling, and filter-based modeling including Wiener and Kalman filters. Radke et al. [6] and Goyette et al. [7] survey a wide range of change detection algorithms related to background subtraction.

The goal of registration and change detection is to align and identify regions of difference between a pair of images. Several issues need to be handled including sensor noise, illumination changes, motion, and atmospheric distortions. The computational challenge lies in the fact that in addition to the modeling operations being complex, the size of these images is also typically large. In this work, we address the twin problem of alignment and differencing in the presence of camera motion. This scenario is unavoidable during long exposures especially when low-lit scenes are being captured. The same would be true if the capturing mechanism itself is moving, for example in drone surveillance systems.

Removal of the static camera constraint brings in a new dimension of complexity to existing change detection methods. One of the main problems that arises is the presence of motion blur since traditional feature-based registration and occlusion detection methods cannot be used due to photometric inconsistencies as pointed out by Yuan et al. [8]. One possible solution is to attempt deblurring as it is possible to obtain a sharp image from the blurred observation through many of the available deblurring methods before sending to the change detection pipeline. However, deblurring invariably introduces artifacts such as ringing which is a hindrance to detecting the actual changes.

Yet another challenge that stems from motion is related to the underlying shutter mechanism itself. In a typical camera using CCD sensors, all pixels are exposed at the same time; these cameras are called global shutter (GS) cameras. They produce motion blur in the captured image when there is camera motion during exposure. However, contemporary CMOS sensors employ an electronic rolling shutter (RS) in which the horizontal rows of the sensor array are scanned at different times. This behaviour results in additional deformations when capturing dynamic scenes and when imaging from moving cameras. Fig. 1 shows the mechanism by which sensors are exposed in RS and global shutter (GS) cameras. A GS camera exposes all the pixels at the same time. Fig. 1(a) illustrates this operation by showing same start and end exposure times for each row of the sensor array. The rows of an RS camera sensor array, on the other hand, are not exposed simultaneously. Instead, the exposure of consecutive rows starts sequentially with a delay as shown in Fig. 1(b).

In reality, it is apparent that both rolling shutter and motion blur issues will be present due to non-negligible exposure time. In the efforts addressed within this proposal, we develop a general model to address both rolling shutter effect and motion blur. Fig. 2 illustrates the four types of distortions that could occur in GS and RS cameras depending on the camera motion. In Fig. 2(a), the GS camera is stationary and the observed image is clean without any distortions. In Fig. 2(b), the camera motion experienced by every row is the same for a GS camera which results



(a) Global shutter (b) Rolling shutter

Figure 1: Exposure mechanism of global shutter and rolling shutter cameras.

in a motion blurred image. In contrast, each row observes a unique camera motion in an RS camera which results in rolling shutter effect and motion blur as shown in Fig. 2(d). The RS effect causes straight lines to appear curved, along with blur. If the camera moves in such a way that each row of the RS camera experiences a single camera pose instead of multiple poses, the motion blur in rows would be negligible as in Fig. 2(c). Yet the RS effect cannot be avoided due to camera motion.

GS GS+MB RS RS+MB

(a) (b) (c) (d)

Figure 2: Various types of distortions in GS and RS cameras.

Previous works on rolling shutter effect [9, 10, 11, 12] model distortions on the image plane whereas we model a general camera motion using projective geometry. Ringaby and Forssén [13] and Grundmann et al. [14] estimate coarse camera motion at specific rows inside a frame, and then interpolate across rows. We instead estimate dense camera motion through all the rows of an image. The above works did not consider the presence of motion blur along

with rolling shutter effect, and they inevitably fail in such a scenario since feature detection is employed. Meilland et al. [15] assume constant camera velocity inside a frame while dealing with combined rolling shutter effect and motion blur. In our work, we deal with both rolling shutter effect and motion blur under a single roof without making any assumptions on camera velocity or on camera motion path parametrization.

In the works described in [16] and [17], change detection was formulated for only planar scenes; [17] deals with only motion blur, while [16] deals with rolling shutter effect and motion blur. In this work, we extend and generalise the image formation model and change detection methodology to 3D scenes in the presence of both rolling shutter and motion blur (RSMB) *aka.* change detection with RSMB distortions.

Our work is the first work of its kind to i) perform registration between a reference image and an image captured at a later time but distorted with *both* rolling shutter and motion blur artifacts, and ii) to also simultaneously detect occlusions in the distorted image, all within a *single* framework. The *scene itself can be 3D* in nature which is what makes this task very challenging. This work is under review with the journal *IEEE Transactions on Pattern Analysis and Machine Intelligence.*

2 Change detection in 3D scenes: Proposed approach

We discretise the model of combined rolling shutter and motion blur with respect to a finite camera pose space $\mathcal{S}$. We assume that the camera can undergo only a finite set of poses during the total exposure time, and this is represented by $\mathcal{S} = \{\boldsymbol{\tau}_k\}_{k=1}^{|\mathcal{S}|}$. This yields the relation

$$\mathbf{g}^{(i)} = \sum_{\boldsymbol{\tau}_k \in \mathcal{S}} \omega_{\boldsymbol{\tau}_k}^{(i)} \, \mathbf{f}_{\boldsymbol{\tau}_k}^{(i)} \tag{1}$$

where $\mathbf{f}_{\boldsymbol{\tau}_k}^{(i)}$ is the ith row of the warped reference image $\mathbf{f}_{\boldsymbol{\tau}_k}$ due to camera pose $\boldsymbol{\tau}_k$. Pose weight $\omega_{\boldsymbol{\tau}_k}^{(i)}$ denotes the fraction of exposure time t_e, that the camera has spent in the pose $\boldsymbol{\tau}_k$ during the exposure of ith row. Since the pose weights represent time, we have $\omega_{\boldsymbol{\tau}_k} \geq 0$ for all $\boldsymbol{\tau}_k$. When the exposure times of $\mathbf{f}^{(i)}$ and $\mathbf{g}^{(i)}$ are same, then by conservation of energy, we have $\sum_{\boldsymbol{\tau}_k \in \mathcal{S}} \omega_{\boldsymbol{\tau}_k}^{(i)} = 1$ for each i.

Our model is general enough that it encompasses both GS and RS camera acquisition mechanisms with and without motion blur (MB). Here $\boldsymbol{\omega}^{(i)}$ is the pose weight vector of the ith row with each of its elements $\omega_{\boldsymbol{\tau}_k}^{(i)}$ representing a number between 0 and 1, which is the weight for the $\boldsymbol{\tau}_k$th pose in the ith row. Each camera pose τ_k is a 6D vector

representing 3D camera translations T_k and 3D camera rotations R_k. The corresponding motion on the image plane is denoted by t_k and r_k.

The image formation model of (1) can be generalized to 3D scenes by following a layered approach. Let us consider L depth layers in the scene and let the set $\mathcal{L} = \{\ell\}_{\ell=1}^{L}$ index the layers. The relative depth of each layer is given by $\{d_\ell\}_{\ell=1}^{L}$, with d_1 is the layer closest to the camera and d_L is the layer farthest from the camera (i.e. the background layer). These values are normalized with respect to the background layer so that $d_L = 1$ and $d_i > d_j$ for $i > j$. During the exposure of the scene, the layer ℓ could possibly be masked by the layers $\{\ell'\}_{\ell'=1}^{k-1}$ at the image plane of the camera. The mask at each layer depends on the homography due to camera pose at that layer since the motion of an object depends on its depth.

Let $\boldsymbol{\alpha}_{(\boldsymbol{\tau}_k,\ell)}$ be the object mask of a layer ℓ at camera pose $\boldsymbol{\tau}_k$. This variable indicates where the objects are present at a particular layer. $\boldsymbol{\alpha}_{(\boldsymbol{\tau}_k,\ell)}$ contains 1 for the pixels where objects are present (i.e. where the layer could possibly contribute to the final image) and 0 otherwise. Let $\boldsymbol{\beta}_{(\boldsymbol{\tau}_k,\ell)}$ denote the final layer mask that indicates the actual contribution of a layer to the final image. We have $\boldsymbol{\beta}_{(\boldsymbol{\tau}_k,\ell)} = \boldsymbol{\alpha}_{(\boldsymbol{\tau}_k,\ell)} \prod_{j=1}^{\ell-1} (\overline{\boldsymbol{\alpha}}_{(\boldsymbol{\tau}_k,j)})$, where $\overline{\boldsymbol{\alpha}}_{(\boldsymbol{\tau}_k,j)}$ is the complement of the object mask indicating the blockage of a layer from being seen at the image plane due to layers in front of it. The above discussion is valid for each row of the observed image. Hence the observed image due to all the camera poses is given by

$$\mathbf{g}^{(i)} = \sum_{\boldsymbol{\tau}_k \in \mathcal{S}} \omega_{\boldsymbol{\tau}_k}^{(i)} \sum_{\ell=1}^{L} \boldsymbol{\beta}_{(\boldsymbol{\tau}_k,\ell)} \widehat{\mathbf{f}}_{(\boldsymbol{\tau}_k,\ell)}^{(i)} \text{for } i = 1, \ldots, M. \tag{2}$$

Here $\widehat{\mathbf{f}}$ represents the complete scene information at each layer. Another way to look at it is $\widehat{\mathbf{f}}_{(\boldsymbol{\tau}_k,\ell)}$ would be the image seen by the camera if it were the only layer present in the scene. Note here that though the camera poses are same for all layers, the actual warp experienced by each layer is different based on its depth. This formation model is readily applicable to a GS image in which the row-wise formulation as in (2) and a global formulation by dropping the superscript (i) are equivalent.

We now represent (2) in terms of images of disjoint layer images. The distorted image can be given by

$$\mathbf{g}^{(i)} = \sum_{\ell=1}^{L} \sum_{\boldsymbol{\tau}_k \in \mathcal{S}} \omega_{\boldsymbol{\tau}_k}^{(i)} \mathbf{f}_{(\boldsymbol{\tau}_k,\ell)}^{(i)} \tag{3}$$

where

$$\mathbf{f}_{(\boldsymbol{\tau}_k,\ell)}^{(i)} = \boldsymbol{\beta}_{(\boldsymbol{\tau}_k,\ell)} \widehat{\mathbf{f}}_{(\boldsymbol{\tau}_k,\ell)}^{(i)} \tag{4}$$

represent disjoint layers for a particular pose $\boldsymbol{\tau}_k$. Note that the clean reference image is given by $\mathbf{f} = \sum_{\ell=1}^{L} \mathbf{f}_{(\boldsymbol{\tau}_k=\mathbf{o},\ell)} = \boldsymbol{\beta}^k_{(\boldsymbol{\tau}_k=\mathbf{o},\ell)} \widehat{\mathbf{f}}_{(\boldsymbol{\tau}_k=\mathbf{o},\ell)}$, where $\mathbf{o}$ represents zero camera motion. Planar scene model in (1) is a special case with $L = 1$, the only layer being the background layer.

2.1 Change Modelling

The relationship (2) between the reference image and the distorted image due to camera motion accounts only for the camera motion and does not account for any change in the scene. We model any new objects or 'changes' in the scene as an additive component. Thus the distorted image is given by

$$\mathbf{g}^{(i)} = \sum_{\ell=1}^{L} \sum_{\boldsymbol{\tau}_k \in \mathcal{S}} \omega^{(i)}_{\boldsymbol{\tau}_k} \mathbf{f}^{(i)}_{(\boldsymbol{\tau}_k,\ell)} + \boldsymbol{\chi}^{(i)} \tag{5}$$

The change vector $\boldsymbol{\chi}^{(i)}$ incorporates changes in all layers. Linear combination of different warps of the image in Eq. (5) can be expressed in matrix-vector multiplication form as

$$\mathbf{g}^{(i)} = \sum_{\ell=1}^{L} \mathbf{F}^{(i)}_{\ell} \boldsymbol{\omega}^{(i)} + \boldsymbol{\chi}^{(i)} \tag{6}$$

where columns of $\mathbf{F}^{(i)}_{\ell}$ contain rows of warped versions of the reference image at layer ℓ and each column gets a weight value from $\boldsymbol{\omega}^{(i)}$. Upon rearranging,

$$\mathbf{g}^{(i)} = [\sum_{\ell=1}^{L} \mathbf{F}^{(i)}_{\ell} \;\; \mathbf{I}][\boldsymbol{\omega}^{(i)} \;\; \boldsymbol{\chi}^{(i)}]^T \tag{7}$$

$$= \mathbf{B}^{(i)} \boldsymbol{\xi}^{(i)} \tag{8}$$

The matrix $\mathbf{B}^{(i)}$ has two parts, the first one being a collection of all warps of the reference image and the second part being the identity matrix to account for the changes. The first part of the vector $\boldsymbol{\xi}^{(i)}$ provides weights to the warps and the second part provides weights to the identity matrix (i.e. the values for the changed pixels). To stack up the columns with warps, the depth map of the scene must be known so that a homography could found out for each depth layer in the scene. If the depth map of the scene is known, we can detect changes in the distorted image by solving for the camera motion and the change vector weights in (8). Since the problem is under-determined, one can formulate the following optimization problem to solve for the camera motion and the changes jointly.

$$E(\boldsymbol{\xi}^{(i)}) = \|\mathbf{g}^{(i)} - \mathbf{B}^{(i)} \boldsymbol{\xi}^{(i)}\|_2^2 + \rho_{\boldsymbol{\omega}}(\boldsymbol{\omega}^{(i)}) + \rho_{\boldsymbol{\chi}}(\boldsymbol{\chi}^{(i)}) \tag{9}$$

The first term imposes the photometric constancy constraint accounting for camera motion and the change, while the second and third terms are priors on camera motion and the change. We observe that (i) camera can move only so much in the whole space of 6D camera poses, and (ii) occlusion is sparse in all rows in spatial domain. The application of sparsity of camera motion is common in literature. We use ℓ_1 norm that encourages sparsity on both the camera motion and the change. We also note that the camera pose weights represent the fraction of time for camera poses and hence we also impose a non-negativity constraint on $\boldsymbol{\omega}^{(i)}$.

$$\min E(\boldsymbol{\xi}^{(i)}) = \|\mathbf{g}^{(i)} - \mathbf{B}^{(i)}\boldsymbol{\xi}^{(i)}\|_2^2 + \lambda_1\|\boldsymbol{\omega}^{(i)}\|_1 + \lambda_2\|\boldsymbol{\chi}^{(i)}\|_1 \text{ with } \boldsymbol{\omega}^{(i)} \succeq \mathbf{0} \tag{10}$$

where λ_1 and λ_2 are non-negative regularisation parameters and $\succeq$ denotes non-negativity of each element of the vector. To enforce different sparsity levels on camera motion and occlusion, we use two ℓ_1 regularisation parameters with different values.

If the depth map of the scene is not known, which is the case in most scenarios, it is not possible to warp different layers according to their depths and stack up the columns of $\mathbf{B}^{(i)}$. Hence, we follow a layered registration approach in which we start by registering the background layer of the distorted image with the reference image by estimating the camera motion and changes, and then registering the change regions one-by-one to other layers, thereby detecting the actual changes as regions which are not registered to any of the layers.

2.1.1 Segmentation of Change Regions

The non-background layers and the changes in the distorted image will be detected by $\boldsymbol{\chi}_K^{(i)*}$. These detected layers will contain many objects at different depths. Some of these objects may be present in the reference image also, and some may not. The objects which are not present in the reference image should be detected as final changes. To mark the regions of changes from $\boldsymbol{\chi}_K^{(i)*}$, we calculate a threshold based on the entropy of the histogram and perform connected component analysis to remove noisy pixels that are smaller than a fixed size.

The segmented regions may not be smooth and continuous. Due to the homogeneous regions present inside the objects, the changes detected may contain holes in them. Before trying to register each object, we need to extract each object separately. We determine the distance transform image of the resulting object layer image. Distance transform assigns a value for each pixel based on the distance of it from the nearest black pixel. We then threshold the distance transform image, and create a binary image. The pixels which have distance transform values lesser than a

preset threshold are assigned a value of one, while others are assigned a value zero. The resultant image will give us a binary image containing all the objects without any holes in them. To extract each region separately, we use watershed segmentation. We find the distance transform of the filled-in object layer image, and segment it using the watershed algorithm. Each object will be numbered including the background layer (black layer). We ignore the background layer using a simple threshold on the ratio of the number of pixels in the layer and the total number of pixels in the image. We name these object regions $\{\widehat{\boldsymbol{\chi}}_p\}_{p=1}^P$ and extract regions containing the objects from the observed image $\mathbf{g}$ as $\{\mathbf{g}_p\}_{p=1}^P = \{\mathbf{g} \cdot \widehat{\boldsymbol{\chi}}_p\}_{p=1}^P$.

2.1.2 Detection of Final Changes

We now aim to register each of these objects $\{\mathbf{g}_p\}_{p=1}^P$ with the reference image. If the registration of an object is not fruitful, then the corresponding region is considered a change. One could try to register each extracted object region from the distorted image with the reference image by estimating the camera motion. We adopt a simpler procedure which uses the fact that the camera motion is same for both the background layer and all other layers and the pose weight vector for all layers remain the same since the exposure period for all layers is same.

A homography pertains only to a plane in the scene. But if we have the rotations and translations observed in the image plane for a particular layer due to camera motion, then we can arrive at the set of rotations and translations observed for another layer. Rotation remains the same irrespective of the depth of the layer while translation scales with respect to depth.

Algorithm 1 Steps for 3D change detection

1: Register background layer of every row using $\{\mathcal{S}^{(i)}\}$ and estimate $\{\boldsymbol{\omega}_L^{(i)*}\}$ and $\{\boldsymbol{\chi}_L^{(i)*}\}$.

2: Extract objects $\{\mathbf{g}_p\}$ from $\{\boldsymbol{\chi}_L^{(i)*}\}$ as in Section 2.1.1.

3: Register each object $\mathbf{g}_p$ using the scaled tuples $\{(\mathcal{S}_\ell^{(i)}, \boldsymbol{\omega}_L^{(i)*})\}$ as in Section 2.1.2.

4: If there is no d_ℓ that registers $\mathbf{g}_p$, mark it as change.

3 Experiments

We demonstrate the working of our registration and change detection method on synthetic as well as real examples including images from the WAMI dataset.

3.1 Synthetic experiments

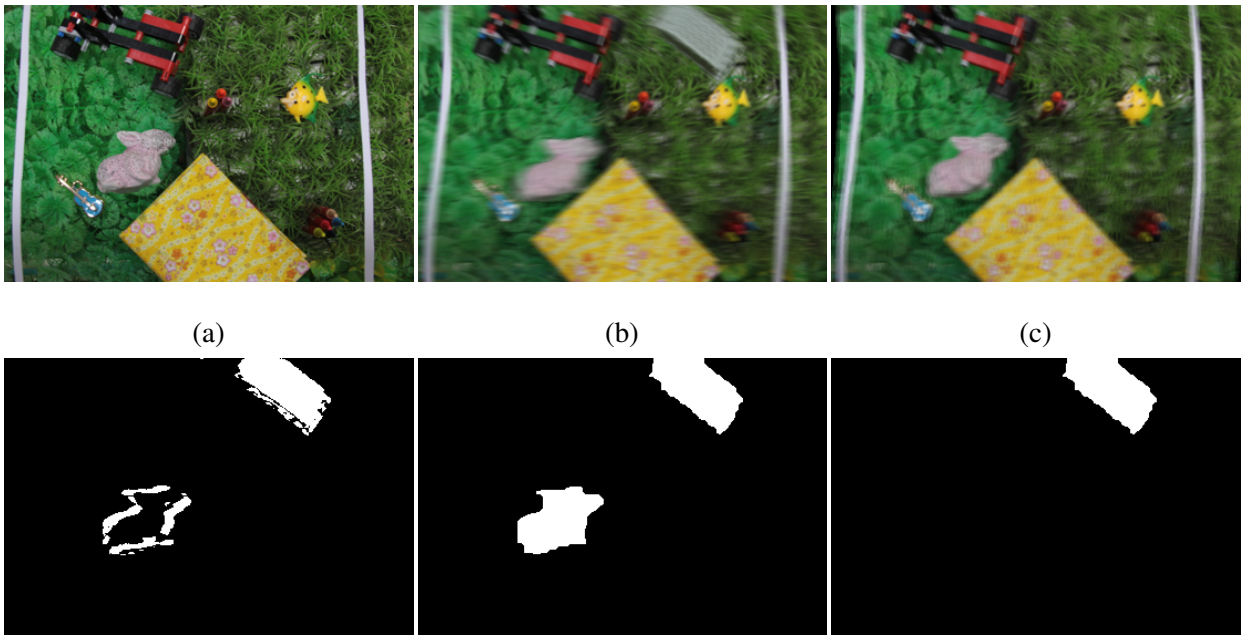
Fig. 3(a) shows a clean image of a 3D scene and Fig. 3(b) shows its corresponding depth map. The darker the region, the farther it is from the camera. The background layer is shown in dark gray since it is farthest from the camera. There are two objects at different depths; one (*bunny*) at a relative depth of 0.5 and another (*block*) at 0.4 with respect to background. The camera motion is applied synthetically on this scene with objects according to the image formation model in (2). For every pose in the camera motion trajectory, the warp is applied on each layer based on its depth, and the warped layers are added. The final image is obtained by the weighted average of warped images of all camera poses. This RSMB image thus generated is shown in Fig. 3(c). The layer with lower relative depth experiences larger motion since it is closer to the camera.

(a) (b) (c)

Figure 3: Synthetic experiment: Change detection in a 3D scene. (a) Reference image, (b) Depth map, and (c) RSMB image.

We follow the steps in Algorithm 1 to detect the changes. To perform the registration of the background layer, we choose a 3D camera pose space as before. We solve for the pose weight vectors for each row. The background-registered image is shown in Fig. 4(c). The change vector weights are non-zero for the unregistered regions, which correspond to both objects, but only one of which is the actual change. This is shown as a binary image in Fig. 4(d). Note that the *bunny* object which is actually present at a different depth is registered except for the borders, since the intensities within the object are mostly homogeneous, but the size of blur along the borders will be different so that the border is clearly marked as change. We fill and extract these binary regions, one at a time, using the method discussed in Section 2.1.1. The filled-in binary object image is shown in Fig. 4(e). For each object, we try to find a relative depth so that it gets registered to the reference image at that particular depth with the same camera motion as that estimated for the background layer. This is done by registering with the scaled tuple as in Section 2.1.2. We vary

the relative depth values from 0.3 to 1.5 in steps of 0.1. The *bunny* object gets registered at a relative depth of 0.5, but the *block* object does not register at all at any depth. Hence, the *block* is marked as a change as shown in Fig. 4(f).



(a) (b) (c)

(d) (e) (f)

Figure 4: Synthetic experiment: Change detection in a 3D scene. *Case-(i)* One-object change. (a) Reference image, (b) RSMB image, (c) Background registered image, (d) Detected objects, (e) Extracted objects, and (f) Detected changes.

3.2 Real Experiments

We next show examples for registration and change detection in real 3D scenes. We capture a scene from the top of a building looking down. The reference image is captured without moving the camera and is shown in Fig. 5(a). The distorted image is captured with predominant horizontal translatory motion of the camera. This can be observed from the shearing effect in Fig. 5(b). This image also has heavy motion blur, and it has two new objects as changes. The majority of the scene is the ground plane and can be considered planar, but the small parapet in the bottom right is at a distance different from that of the ground, and hence it incurs a different amount of blur and rolling shutter effect. We first register the background plane as shown in Fig. 5(c). The ground gets registered with correct motion estimation, but the parapet does not, which is as expected. This can be seen from the border of the parapet in the thresholded

image in 5(d). At this stage, the actual changes have also been correctly detected. We extract each object by filling in the holes following the procedure in Section 2.1.1. For each object in Fig. 5(e), we find a scale so that it gets registered at a particular relative depth following Algorithm 1. The parapet gets registered at a relative depth of 0.8. The other two objects are not registered and hence they are considered as the final changes, which are shown in Fig. 5(f).

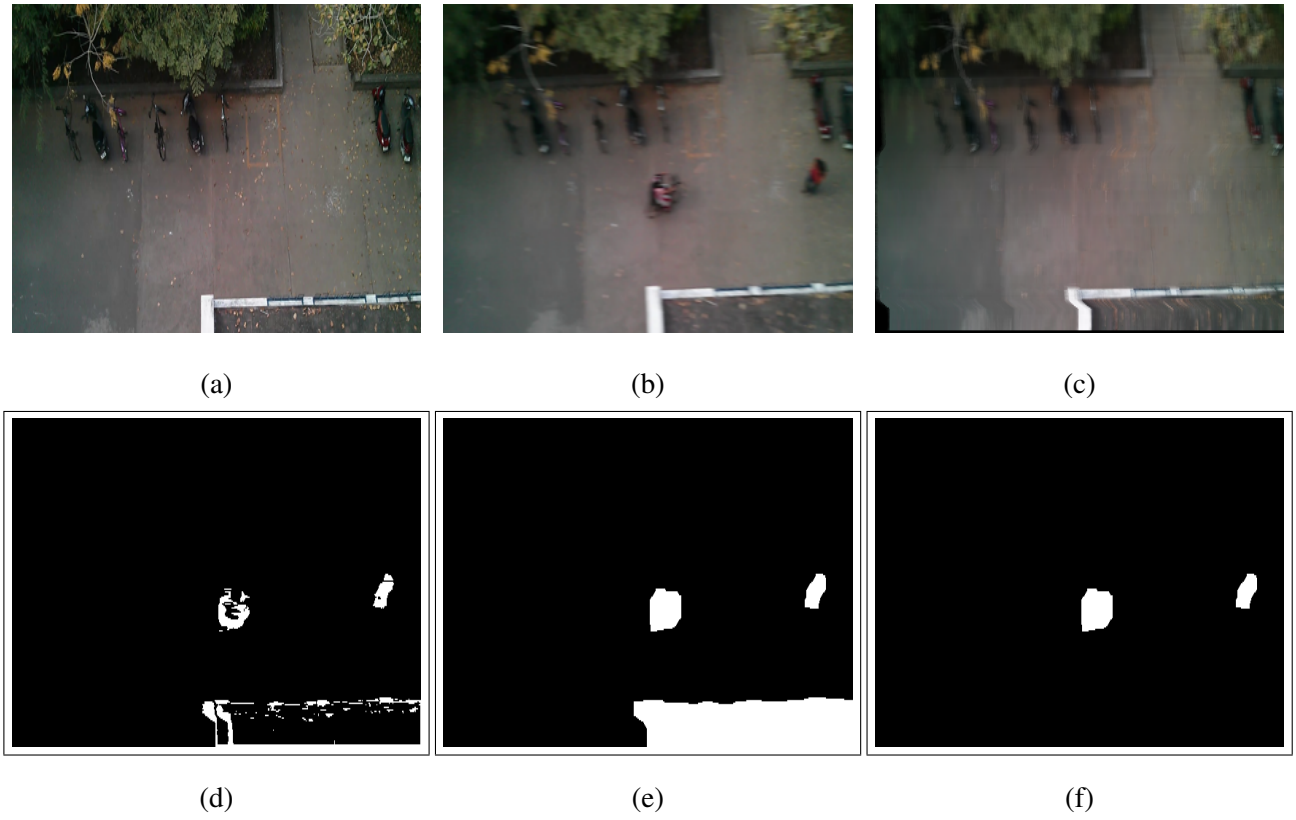
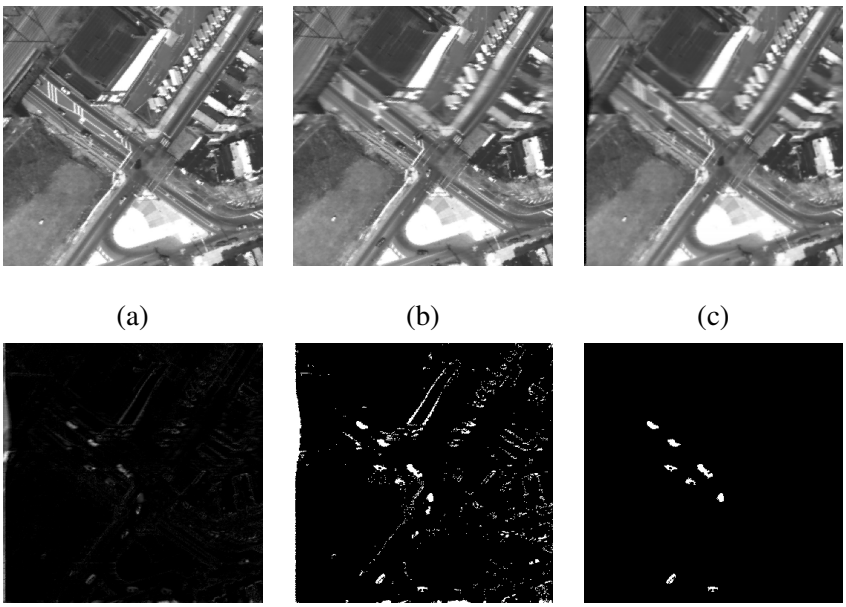


Figure 5: Real experiment: Change detection in a 3D scene. (a) Reference image, (b) RSMB image, (c) Background-registered image (d) Detected objects, (e) Extracted objects, and (f) Detected changes.

We also compare our result with of Liang [10] and Ringaby [13]. We estimate the camera motion using Figs. 5(a) and (b) for these two methods. We then follow the distort-difference pipeline by applying the camera motion on the reference image and detecting the changes. The final changes detected bythe methods of Liang and Ringaby are shown in Figs. 6(b) and (c) respectively. Our result is shown again in Fig. 6(a) for comparison. The competing methods are applicable only for planar scenes and hence there are misregistrations in the boundary of the parapet. Even otherwise, due to the presence of motion blur, both these methods have misregistrations across the image since they cannot handle blur. Ringaby estimates the camera motion better than Liang but our method yields best results.

(a) Ours (b) Liang (c) Ringaby

Figure 6: Comparisons with existing methods for RSMB change detection in a 3D scene. Detected changes between Figs. 5(a) and (b) using (a) our method, (b) Liang, and (c) Ringaby.



(a) (b) (c)

(d) (e) (f)

Figure 7: (a) Reference image, (b) Distorted images with occlusions, (c) Registered image, (d) Grayscale occlusion image, (e) Binary occlusion image, and (f) Detected changes.

Next, we show representative examples for registration and change detection on the WAMI dataset which contains aerial data captured from a moving platform. Figs. 7(a) and (b) show the reference and distorted images between which changes are to be detected. There are local blur distortions in Fig. 7(b) which can be observed from the curved road in the top-right quandrant. We need to register the two images for camera motion, and in addition, we need to account for the occlusion objects. For each row, we estimate the camera motion using a joint row-wise registration

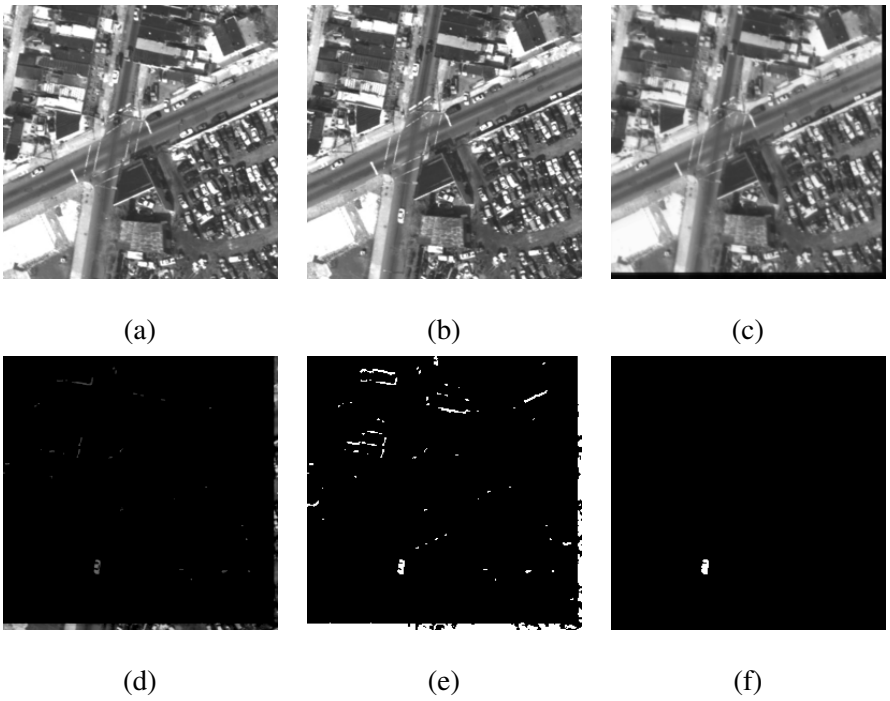


Figure 8: (a) Reference image, (b) Distorted images with occlusions, (c) Registered image, (d) Grayscale occlusion image, (e) Binary occlusion image, and (f) Detected changes.

and occlusion detection framework. The registered image is shown in Fig. 7(c). The occlusion grayscale image is shown in Fig. 7(d). The occlusion binary image after thresholding is shown in Fig. 7(e). Due to the 3D nature of the scene, the edges of the buildings are also detected as occlusions. This is because the buildings are present at a different depth compared to the ground. We remove the false positives using a connected component approach, where we remove the thin edges of the buildings and other noise from the occlusion image. The final changes are shown in Fig. 7(f) and correctly correspond to actual changes in the scene berween the two frames.

An example with global motion in a 3D scene in shown in Fig. 8. The reference and distorted images are shown in Figs. 8(a) and (b), respectively. The registered and occlusion images are shown in Figs. 8(c) and (d), respectively. The thresholded occlusion image is shown in Fig. 8(e). After removing the changes due to depth varitions and noise, yet again, our algorithm detects changes correctly as shown in Fig. 8(f).

4 Conclusions

Registration and change detection are important preprocessing tasks in many image processing and computer vision applications. A wide variety of challenges exist within the change detection paradigm, including camera motion, shutter mechanism, illumination changes, and depth variations.

In this work, we considered the two-image change detection problem in which one of the images is affected by rolling shutter effect and motion blur. We proposed a new method to jointly estimate the camera motion that had occurred during the exposure period and the region of changes. We formulated a general model which covers image acquisition using both global shutter and rolling shutter cameras. We also considered the presence of 3D objects in the distorted image, which our model successfully registered to the reference image or detected as a change true to the nature of the object. Our method was tested on many examples both synthetic and real to demonstrate its effectiveness.

Publications arising out of efforts of Year-1

1. Vijay Rengarajan, Abhijith Punnappurath, A. N. Rajagopalan, and Guna Seetharaman. Efficient Change Detection for Very Large Motion Blurred Images. in *IEEE Computer Vision and Pattern Recognition Workshop on Registration of Very Large Images*, pp. 315-322, 2014.

2. Vijay Rengarajan, Sheetal B. Gupta, A.N. Rajagopalan, and Guna Seetharaman. Illumination Robust Change Detection with CMOS Imaging Sensors. *Proc. SPIE Defense + Security*, 9473, pp. 1-9, 2015.

3. Arun M, A.N. Rajagopalan and Gunasekaran Seetharaman. Multi-Shot Deblurring for 3D Scenes. in *IEEE International Conference on Computer Vision Workshop on Inverse Rendering*, 2015 (accepted).

4. Vijay Rengarajan, A.N. Rajagopalan, Aravind Rangarajan and Guna Seetharaman. Change Detection of Motion Blurred 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (under review).

5. Vijay Rengarajan, A.N. Rajagopalan, Aravind Rangarajan and Guna Seetharaman. Rolling Shutter: Rectification and Mosaicing. *IEEE Transactions on Image Processing* (to be submitted).

References

[1] R. Feris, J. Petterson, B. Siddiquie, L. Brown, and S. Pankanti, "Large-scale vehicle detection in challenging urban surveillance environments," in *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, Jan 2011, pp. 527–533. 1

[2] S. Sivaraman and M. Trivedi, "A general active-learning framework for on-road vehicle recognition and tracking," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 2, pp. 267–276, June 2010. 1

[3] C.-Y. Fang, S.-W. Chen, and C.-S. Fuh, "Automatic change detection of driving environments in a vision-based driver assistance system," *Neural Networks, IEEE Transactions on*, vol. 14, no. 3, pp. 646–657, May 2003. 1

[4] L. Bruzzone and F. Bovolo, "A novel framework for the design of change-detection systems for very-high-resolution remote sensing images," *Proceedings of the IEEE*, vol. 101, no. 3, pp. 609–630, March 2013. 1

[5] T. Bouwmans, "Recent advanced statistical background modeling for foreground detection-a systematic survey," *Recent Patents on Computer Science*, vol. 4, no. 3, pp. 147–176, 2011. 1

[6] R. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, "Image change detection algorithms: a systematic survey," *Image Processing, IEEE Transactions on*, vol. 14, no. 3, pp. 294–307, March 2005. 1

[7] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "A novel video dataset for change detection benchmarking," *Image Processing, IEEE Transactions on*, vol. 23, no. 11, pp. 4663–4679, Nov 2014. 1

[8] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum, "Blurred/non-blurred image alignment using sparseness prior," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, Oct 2007, pp. 1–8. 1

[9] O. Ait-Aider, N. Andreff, J. Lavest, and P. Martinet, "Simultaneous object pose and velocity computation using a single view from a rolling shutter camera," in *Computer Vision ECCV 2006*, ser. Lecture Notes in Computer Science, A. Leonardis, H. Bischof, and A. Pinz, Eds. Springer Berlin Heidelberg, 2006, vol. 3952, pp. 56–68. [Online]. Available: http://dx.doi.org/10.1007/11744047_5 1

[10] C.-K. Liang, L.-W. Chang, and H. Chen, "Analysis and compensation of rolling shutter effect," *Image Processing, IEEE Transactions on*, vol. 17, no. 8, pp. 1323–1330, Aug 2008. 1, 3.2

[11] W.-H. Cho, D.-W. Kim, and K.-S. Hong, "Cmos digital image stabilization," *Consumer Electronics, IEEE Transactions on*, vol. 53, no. 3, pp. 979–986, Aug 2007. 1

[12] S. Baker, E. Bennett, S. B. Kang, and R. Szeliski, "Removing rolling shutter wobble," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, June 2010, pp. 2392–2399. 1

[13] E. Ringaby and P.-E. Forssén, "Efficient video rectification and stabilisation for cell-phones," *International Journal of Computer Vision*, vol. 96, no. 3, pp. 335–352, 2012. [Online]. Available: http://dx.doi.org/10.1007/s11263-011-0465-8 1, 3.2

[14] M. Grundmann, V. Kwatra, D. Castro, and I. Essa, "Calibration-free rolling shutter removal," in *Computational Photography (ICCP), 2012 IEEE International Conference on*, April 2012, pp. 1–8. 1

[15] M. Meilland, T. Drummond, and A. Comport, "A unified rolling shutter and motion blur model for 3d visual registration," in *Computer Vision (ICCV), 2013 IEEE International Conference on*, Dec 2013, pp. 2016–2023. 1

[16] V. Rengarajan, A. Rajagopalan, and A. Rangarajan, "Change detection in the presence of motion blur and rolling shutter effect," in *Computer Vision - ECCV 2014*, ser. LNCS. Springer, 2014, vol. 8695, pp. 123–137. 1

[17] V. Rengarajan, A. Punnappurath, A. Rajagopalan, and G. Seetharaman, "Efficient change detection for very large motion blurred images," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, June 2014, pp. 315–322. 1